

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Service Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.</p>						
1. REPORT DATE (DD-MM-YYYY) 11/09/2012		2. REPORT TYPE Final Report			3. DATES COVERED (From - To) Aug 2009-Aug 2012	
4. TITLE AND SUBTITLE MULTISCALE MODELING AND SIMULATION OF COMPRESSIBLE MULTIMATERIAL FLOW				5a. CONTRACT NUMBER FA9550-09-1-0146		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) H. S. Udaykumar				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The University of Iowa 2408 Seamans Center, Iowa City, IA-52242					8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Computational Mathematics Program, AFOSR					10. SPONSOR/MONITOR'S ACRONYM(S)	
					11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-OSR-VA-TR-2012-1066	
12. DISTRIBUTION/AVAILABILITY STATEMENT Distribution A - Approved for Public Release						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT <p>The evolution of a solid-gas mixture under the influence of a shock wave depends on particle-particle and particle-shock interactions; i.e. the macroscopic distribution of particles is subject to physics at the particle-scale. This work seeks to simulate the macro-scale dynamics of gas-solid mixtures by employing information accumulated from direct numerical simulations (DNS) at the micro-scale. Data on the forces experienced by particles in a cloud are collected from DNS using a compressible Eulerian solver and provided to an artificial neural network (ANN); the simulations are performed for a range of control parameters, such as Mach number, particle radii, particle-fluid density ratio, position, and volume fraction. Beginning with a simple single stationary particle case and progressing to moving particle laden clouds, the ANN is trained to evolve and reproduce correlations between the control parameters and particle dynamics. The trained ANN is then used in computing the macro-scale flow behavior in a model of shocked dusty gas advection. The model predicts particle motion and other macro-scale phenomena in agreement with experimental observations.</p>						
15. SUBJECT TERMS Multiscale modeling, multimaterial flows, compressible flows, shocks						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 38	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code)	

FINAL REPORT FOR GRANT # FA9550-09-1-0146

MULTI-SCALE MODELING AND SIMULATION OF COMPRESSIBLE MULTI-MATERIAL FLOWS

PI: H. S. UDAYKUMAR

Professor, Department of Mechanical and Industrial Engineering

The University of Iowa, Iowa City, IA-52242

1. ABSTRACT

The evolution of a solid-gas mixture under the influence of a shock wave depends on particle-particle and particle-shock interactions; i.e. the macroscopic distribution of particles is subject to physics at the particle-scale. This work seeks to simulate the macro-scale dynamics of gas-solid mixtures by employing information accumulated from direct numerical simulations (DNS) at the micro-scale. Data on the forces experienced by particles in a cloud are collected from DNS using a compressible Eulerian solver and provided to an artificial neural network (ANN); the simulations are performed for a range of control parameters, such as Mach number, particle radii, particle-fluid density ratio, position, and volume fraction. Beginning with a simple single stationary particle case and progressing to moving particle laden clouds, the ANN is trained to evolve and reproduce correlations between the control parameters and particle dynamics. The trained ANN is then used in computing the macro-scale flow behavior in a model of shocked dusty gas advection. The model predicts particle motion and other macro-scale phenomena in agreement with experimental observations.

2. INTRODUCTION

Phenomena involving high-speed multiphase flows occur in dust explosions, condensation shocks, explosive debris transport, detonation in heterogeneous media and so on. In these flows complex interactions occur between the various coexisting phases, including carrier fluid-particle interactions and particle-particle interactions[1-2]. Such flows are difficult to visualize (due to the wide range of length scales and short time scales involved); experimental measurements are difficult and expensive to obtain. Even where experimental data are available, yielding empirical correlations that encapsulate behavior (e.g. drag laws) the modeling of the mixture dynamics can lead to loss of important physics, i.e. the fine-scale behavior may be homogenized and diffused. Preserving simplicity of the closure model (which transmits fine-scale behavior to the coarse-scale) can exact a toll on the extent to which fine-scale physics is captured at the coarse-scale.

As an archetype of compressible flows of mixtures, computational modeling of shocked particle-laden flows has received much attention. However, in such simulations, one must rely on empirical models to describe the dynamics of the particle phase; in particular empirical drag laws

are employed in effecting particle motions in both Lagrangian and Eulerian treatment of the solid phase. Since the length scales of the discrete particles in a multi-material system and the time scales of response of the particulate phases may be vastly different from that of the bulk flow, resolving the dynamics of the individual components of the mixture is impossible. Therefore some overall (averaged or homogenized) behavior of the multi-material mixture needs to be modeled and computed, and resorting to empiricism is unavoidable. While such averaged material representations may be sufficient for many engineering applications, there are some physical problems where the local behavior of the material, i.e. the detailed interactions between the (unresolved) individual phases in the mixture can become important and can influence the observed global dynamics.

An example of macroscale phenomena that reflect particle-scale dynamics can be seen in the excellent experiments of Boiko et al [1]. In their experiments a cloud of particles (polystyrene, average particle diameter d_p of 80 microns) is hit by a shock wave (traveling from left to right). The overall behavior of the particles subjected to the shock is very interesting; in particular, for the high particle volume fraction case the particle distribution assumes a triangular form as illustrated in Figure 1, while the low particle volume fraction case does not produce a distinct structure. Boiko et al also produced a column of particles in a shock tube and studied the evolution of the column and its interaction with a planar shock. Figure 1 illustrates the response of a column of particles to the shock. In each case, the geometry of the initial particle distribution as well as the volume fraction of the initial cloud determines the macro-scale distribution of the particles following interaction with the shock. For example, the formation of the triangular structure in the case of the heavily loaded gas-solid mixture must hinge upon the interactions between the more densely packed particles; the physics behind the formation of a triangular pattern is recovered by the ANN-based multiscale modeling scheme developed herein and is explained later in this paper.

The particle motions in a macro-scale particle-fluid mixture model traditionally follow from Newton's laws applied to the individual particles and reflect the force transmitted to the individual particles by the impinging shock [2-5]. This force will depend on the shock strength (Mach number, M), the density of the particle relative to the fluid ($\frac{\rho_p}{\rho_f}$), the volume fraction of the solid (ϕ_p) and the particle size (d_p). The key question is: how does one determine the dependency of the force on a given particle on each of these parameters?

The route pursued in this work is to perform direct numerical simulations (i.e. *in silico* experiments) on small clusters of particles subject to a range of conditions in the parameter space defined above (consisting of M , $\frac{\rho_p}{\rho_f}$, ϕ_p , d_p) to learn about the behavior of "representative particles". For example, one can compute the drag versus time curves for particles based on such simulations as a function of the above four parameters. Then one can encapsulate the dependence of the drag on time as well as on the parameters in the form: $D(t) = f(M, \frac{\rho_p}{\rho_f}, \phi_p, d_p, t)$, which is conventionally the route taken in establishing experimental correlations or drag laws. However, since the drag law to be derived is dependent in a rather complex way on multiple parameters, the

resulting manifold in the parameter space that describes the drag law can be quite difficult to obtain. Therefore, the idea of employing a device to “learn” this law from a series of computational experiments becomes attractive. The general concept of utilizing neural architectures to learn behaviors at a given scale that can be transmitted to other scales opens the possibility of using artificial neural networks (ANNs) [6-8] for multiscale modeling. The current approach follows the route of ANN-based learning to effect inter-scale communication, which has been applied in a few instances of multiscale modeling thus far [9-14].

A particular application of artificial intelligence which closely parallels the application herein is that of pattern recognition or knowledge assimilation; this feature has been adopted for use in a variety of fluid dynamics applications [12-13, 15-16]. An ANN is capable of learning complicated behavior, i.e. effectively building a representation of functions of several variables by modifying a collection of weights attached to its “neurons”[8, 18]. The computational effort in ANN applications comes from the need to train the ANN by providing it with sufficient samples of training data, so that the ANN can adequately construct the manifold (in a specified multidimensional parameter space) representing the behavior of the system. The number of samples required to train the ANN depends on the complexity of the behavior to be represented and also depends on the complexity of the ANN itself. Once the ANN is trained however, knowledge recovery is rather rapid, and is performed by interrogating the ANN. This work will seek to demonstrate these concepts by applying it to solve the problem of shock-impacted particle laden flows as pictured in Figure 1.

3. NUMERICS AND CALCULATIONS

3.1 COMPUTATIONAL SET-UP

The micro-scale calculations are in the category of DNS, i.e. they are highly resolved. The computational setup for such simulations would require a domain large enough to contain the incident shockwave, the cloud of particles, bow shocks, and shock reflections. However, the grid size would need to be small enough to capture necessary details of shock-particle interaction, particle motion, shock wave dynamics, transient forces, and sharp interfaces.

Current drag laws for supersonic flow were obtained through physical experiments [2, 25-29]. Most previous work has resorted to using drag laws as functions of Reynolds and Mach numbers. These types of drag laws do not explicitly define unsteady drag but rather an overall drag coefficient once the shock has already passed over the particles. In fact, for small enough particles (i.e. in the micron-range), shock passage is rapid enough that viscous effects can be neglected and the Euler equations can be employed to predict forces on the particles; then, viscous effects come into play at much longer time scales. The inertial time scale can be estimated as:

$$\tau_{inertial} = \frac{d_p}{U_\infty} = \frac{d_p}{a} * \frac{a}{U_\infty} = \frac{d_p}{a} * \frac{1}{M}$$

and the viscous time scale as:

$$\tau_{viscous} = \frac{d_p^2}{\nu} = \frac{d_p}{U_\infty} * \frac{d_p U_\infty}{\nu} = \frac{d_p}{U_\infty} * Re$$

The ratio between the inertial and viscous time scale is:

$$\frac{\tau_{inertial}}{\tau_{viscous}} = \left(\frac{d_p}{a} \frac{1}{M} \right) * \left(\frac{U_\infty}{d_p} \frac{1}{Re} \right) = \left(\frac{U_\infty}{a} \frac{1}{M} \right) * \left(\frac{1}{Re} \right) = Re^{-1}$$

where d_p is the particle diameter, U_∞ is the flow velocity, a is the speed of sound, M is the Mach number, ν is the kinematic viscosity, and Re is the Reynolds number. The Reynolds number is defined as the ratio of inertial forces to viscous forces. For high speed compressible flows, the Reynolds number is very large. It usually lies in the range of 10^5 to 10^6 even for small particles. The implication is that the effects of the viscosity of a fluid would not be significant until the shock is already 10^5 to 10^6 particle diameters away; thus in determining the motion of particles in the instants following shock impingement viscosity may be neglected and the driving force behind shocked particle motion is mainly inertial drag from the shock wave.

For the purpose of making comparisons, our simulations were kept fairly close to numerical calculations[4, 31-35] and experiments performed [1, 19, 27-28, 32, 36-39] and published by others. As mentioned before the parameter space is defined by the Mach number, the particle volume fraction, the relative density of the particle to the fluid and time. Mach numbers were set between 1.2 and 4.0, $\frac{\rho_p}{\rho_f}$ was kept between 100 and 3100, and ϕ_p between 2.0% and 22.4% when large particle arrays were used. For larger particle arrays the setup is similar to the 41 particle cases; whose setup is seen in Figure 14. The shock wave was placed at 5 units from the left wall and traveled to the right.

Because the physics of the problem certain assumptions can be made to simplify the problem without sacrificing accuracy of results. The forces of gravity are negligible; the weight of each particle and movement affected by gravity and buoyancy are neglected in comparison to the drag forces. The fluid phase behaves as an ideal gas; the equation of state is the same as the ideal gas law. The gas and particles are calorically perfect; the specific heat values are constant for both phases. The solid particles are perfectly rigid; they undergo no deformation. There are no collisions; simulations stop when particle level-sets come in contact. In the macro-scale Lagrangian advection, particles are treated as points and may overlap. Thermal boundary layers do not develop in the time frame of shock-particle interaction; therefore adiabatic particle surfaces are assumed, thermal conductivity is set to zero. Kinetic boundary layers do not develop in the time domain; the model is inviscid, dynamic viscosity is ignored, no particle rotation occurs.

3.3 GOVERNING EQUATIONS

The method used solved a set of a governing set of hyperbolic equations for compressible fluid flow[40]. These governing equations when simplified and placed in conservation form in Cartesian coordinates are:

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} = \vec{S} \quad \text{Equation 1}$$

where,

$$\vec{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad \vec{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho E + p) \end{pmatrix}, \quad \vec{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(\rho E + p) \end{pmatrix} \quad \text{and} \quad \vec{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(\rho E + p) \end{pmatrix}$$

In the equations above,

$$E = e + \frac{1}{2}(u^2 + v^2 + w^2) \quad \text{Equation 2}$$

where E is the total internal energy and e is the specific internal energy. For the Euler equations in Cartesian coordinates, the source term \vec{S} , is set to zero. Closure for the governing equations can be achieved by utilizing a stiffened equation of state,

$$P = \rho e(\gamma - 1) - \gamma P_\infty \quad \text{Equation 3}$$

where γ is the specific heat ratio and P_∞ is a material dependent constant. Under the assumption of an ideal gas, we would then have $P_\infty = 0$ and $\gamma = c_p/c_v$.

For stiff fluids such as water, the specific heat ratio and the material dependent constant would assume the values of 5.5 GPa and 6.13 GPa, respectively. Lastly, from the definition of the speed of sound and using the stiffened equation of state, the speed of sound can be calculated by

$$c = \sqrt{\frac{\gamma(P + P_\infty)}{\rho}} \quad \text{Equation 4}$$

3.4 IMMERSED BOUNDARY METHOD

For the consideration of boundary conditions at an interface, an immersed boundary method is used. The algorithm used is an Eulerian-Lagrangian algorithm for interface tracking in three dimensions, otherwise known as ELAFINT3D. The ELAFINT3D code utilizes a sharp interface treatment method as described by Sambasivan [40]. The sharp interface treatment requires continuous tracking and representation for the interface surface. To represent the embedded interface surfaces, Level-sets were used, first introduced by Osher and Sethian[41]. The level-set is simply an intersection between a defined level-set field and the working plane. The level-set field is advected using the level-set advection equation:

$$\frac{\partial \phi_l}{\partial t} = \vec{V}_l \cdot \vec{\nabla} \phi_l = 0 \quad \text{Equation 5}$$

where ϕ_l represents the level-set and \vec{V}_l represents the level-set velocity field for the l^{th} embedded surface. For the solution methodology, a fourth-order essentially non-oscillatory scheme for was used for spatial discretization and a fourth order Runge-Kutta time integration was used to solve the level-set advection equation. The value of the level-set field at ϕ_l any point is the signed normal distance from the l^{th} interface with $\phi_l \leq 0$ inside the immersed boundary and $\phi_l \geq 0$ outside. The interface is implicitly determined by the zero level-set field defined when $\phi_l = 0$, and where the contours represent the l^{th} immersed boundary.

3.5 BOUNDARY CONDITIONS

To handle the jumps in the mass, momentum and energy fluxes along with the material properties across the interface, the tracked interface will have to be coupled with the flow solver to insure an accurate depiction. In the ghost fluid method, this translates to suitably populating the number of ghost points [40, 42-43]. At the interface of a solid body immersed in a compressible flow, the following boundary conditions were applied for velocity, temperature and pressure fields. For no-penetration for normal velocity:

$$v_n = U_n \quad \text{Equation 6}$$

where U_n is the center of mass velocity for the embedded rigid object. To satisfy the slip condition for the tangential velocity:

$$\frac{\partial v_{t_1}}{\partial n} = 0 \quad \text{and} \quad \frac{\partial v_{t_2}}{\partial n} = 0 \quad \text{Equation 7}$$

To satisfy the adiabatic temperature condition:

$$\frac{\partial T}{\partial n} = 0 \quad \text{Equation 8}$$

To keep the normal force pressure balance:

$$\frac{\partial p}{\partial n} = \frac{\rho_s v_{t_1}^2}{R} - \rho_s a_n \quad \text{Equation 9}$$

and

$$v_n = \vec{V} \cdot \hat{n}, \quad v_{t_1} = \vec{V} \cdot \hat{t}_1, \quad v_{t_2} = \vec{V} \cdot \hat{t}_2 \quad \text{Equation 10}$$

where v_n is the normal velocity, v_t is the tangential velocity in the interface referenced curvilinear coordinate, \vec{V} is the velocity vector in the global Cartesian coordinate, \hat{n} , \hat{t}_1 , \hat{t}_2 are the normal and tangential vectors, R is the radius of curvature and a_n is the acceleration of the interface; the set of boundary conditions that govern the behavior of the flow near the embedded solid body and must be enforced on the real fluid by suitably populating the corresponding ghost points[40].

3.6 ARTIFICIAL NEURAL NETWORK

The neural network used is a single hidden layer, feed-forward, back-propagation network[6]. It possesses one hidden layer of neurons between the input layer and output layer. The input layer includes one bias neuron to facilitate different levels of activation for each hidden neuron. The last layer consists of outputs where a final prediction can be used to find an error in the prediction and adapt the weights to the previous layers allowing the ANN to learn. The basic network topography is show in Figure 7.

The ANN must go through two important phases before it will be capable of producing useful predictions. The first phase is the training phase where a set of data is provided and the ANN learns from the data. The algorithm used to learn and edit the weights for each neuron is called a back-propagation algorithm. Every neuron in the network contains the same basis function for processing data. For most cases, there is only one output neuron that sums all its inputs to arrive at a final prediction. A back-propagation algorithm[6] takes the predicted values and compares it to the expected values (i.e. to the target output for the given inputs in the training set). Depending on the error between the two, the weights for each neuron is edited. The testing of the neural network is performed by making a random selection from the data set (until all the data are run through) and each data point is tested and used to train the neural network once per cycle. When the ANN is in training, it should be learning from every point in a data set, otherwise learning will be biased. Every iteration step for an ANN consists of cycling through the total number of data points in a data set. The error produced on every iteration step can be plotted to show a convergence curve on how the ANN is being trained. One such convergence curve for the training of ANN is shown in Figure 8. Note that as the iterations increase the learning of the ANN saturates and convergence is declared at a pre-specified error tolerance or maximum iteration count.

When the training phase is complete, an artificial neural network can be tested by querying with a testing set of input data. The resulting output from the ANN is compared against the desired output corresponding to the input parameters for that testing set. The ANN is assessed to have successfully learned if the error produced for the testing set is below a desired tolerance. Querying an ANN at multiple points inside the parameter space allows testing for the robustness of the prediction from the ANN; in general the prediction deteriorates at the fringes of the parameter space or in regions of parameter space where training data are sparse. The performance of the ANN as a function approximation device is illustrated with some examples below.

4. RESULTS

EXAMPLES OF ANN LEARNING PROCESS

LEARNING A DRAG LAW

When a planar shock wave hits a stationary spherical particle and passes over it, the drag force on the particle (i.e. force exerted on the particle) changes throughout shock passage. Once such drag versus time curve obtained by Tanno et al. [19] in an experimental (shock tube) setup is displayed in Figure 12.

EMPIRICAL DRAG LAWS

Empirical drag laws do not provide the transient drag experienced by the particle as the shock passes over it. Instead, some measure of steady drag is available that omits the details of the shock passage. With trained ANNs, however, one can retain the information on the drag versus time for a wide range of parameter space. Thus, information obtained from experiments or computations need not be discarded; it can be learned and retained as “knowledge” by the ANN[17]. This does not imply that a large data set is stored. Once the ANN is trained the information on the drag versus time behavior is stored in the weights attached to the individual neurons in the ANN; the individual data sets used for training can then be discarded.

“LIFTING” INFORMATION FROM MESO-SCALE CALCULATIONS

The driving force behind particle motion in shock-impacted particle-laden flows is the drag force produced on the particle. Once a shock wave has passed over a particle, the subsequent trajectory of the particle can be determined from Newton’s law if the impulse provided to the particle by the shock is known. To model a particle’s trajectory at the macro-scale, information must be “lifted” from the meso-scale. To limit the amount of information passage between scales, only the most pertinent data is passed. A particle’s position, trajectory and velocity are dependent only on the initial location, mass and force applied. Since the force is transient in nature, its characteristics must be quantified. When viewing a shocked particle drag curve (Figure 12), it is evident that there is a maximum value of force that is reached as the shock impinges on the particle and the drag force decays over time. These two values are maximum drag coefficient, $C_{d_{max}}$ and relaxation time, τ_r . Once the drag versus time curve is established and the $C_{d_{max}}$ and τ_r is known, the total impulse delivered by the shock, I_t , can be computed as the area under the curve. For a standard drag curve (obtained from experiment or simulation), we can set τ_r to be represented by exponential decay and thus the impulse would be:

$$I_t = \int_{t_o}^{t_f} C_{d_{max}} * e^{-t/\tau_r} \text{ Equation 11}$$

where I_t is the impulse, t_o is the impact time, t_f is the final time, $C_{d_{max}}$ is the maximum drag force, t is time, and τ_r is the relaxation time. It turns out that in macro-scale calculations, the quantity of interest is I_t . In addition, since the impulse I_t acts over a time characterized by τ_r , once these two values are known, the momentum change of a particle hit by a shock can be calculated. These two pieces of information are all that is needed to quantify a particle’s trajectory in a macro-scale calculation. Thus, the ANN can be trained to learn these two quantities as functions of the input parameters.

MACRO-SCALE CALCULATIONS

Since the main idea behind using an ANN-based learning scheme was to create an “equation-free” lifting scheme[20-21], macro-scale calculations can employ the information obtained from the ANN in effecting Lagrangian particle motion. Given the Mach number, $\frac{\rho_p}{\rho_f}$, and d_p , an ANN can predict $C_{d_{max}}$ and τ_r . These values can then be placed in a Lagrangian algorithm using Newton’s second law and the particle trajectory calculated.

VERIFICATION

To insure the reliability of our code, the computed drag force obtained was non-dimensionalized using the same parameters as Drikakis et al.[31]. The comparison of the non-dimensional drag force is shown in Figure 15. A visual comparison between the results obtained from the present approach and that of Drikakis et al. is shown in Figure 16 using isodensity lines. The transient drag curves produced by Drikakis et al. and those produced by the present calculations show minimal difference in peak magnitude and are rather similar, even though Drikakis et al. employed Navier-Stokes computations for rather modest Reynolds numbers for their calculations. The similarity of the drag behavior for the Euler and Navier-Stokes computations supports the present inviscid computations for the shock-particle interaction, particularly for the high Reynolds numbers that apply to the particles considered by Boiko et al and targeted in the present work.

SINGLE PARTICLE CASES

The ELAFINT3D code was first used to test a cylindrical particle in a fluid flow during varying conditions. This included experiments of post-shocked flow, a shocked stationary particle, and shocked moving particle. Later on, cases of shocked particle arrays with large number of particles were examined. The single particle tests were set up to illustrate the evolution of data processing the ANN needed to learn in an order of increasing complexity.

STATIONARY PARTICLE

In this case the particle was held stationary and then hit with a shock. The boundary conditions were set the same as the post-shocked particle case except the lower wall set as symmetry. A grid domain of 500 by 250 cells was used for the drag curves calculated from the ELAFINT3D code. This was to match and verify the results by the ELAFINT3D code to those of Drikakis[31] as seen previously. The initial starting distance for the shock wave was set more than the radius of the cylinder away from the cylinder itself. The shock was allowed to impact the cylinder and continue to travel as data for horizontal force was recorded over time. A Schlieren image of the one of the cases is shown in Figure 20.

With a smaller domain size, it would be reasonable to test the effect of grid size and the use of local mesh refinement. For the fine grid, the number of grid cells was increased by four times with the grid sizes half the original. For the local mesh refinement two levels of refinement were used to provide grid cells near the interface with edges a fourth of the original. It was discovered that both the finer grid structure and the use of local mesh refinement show some differences. The differences were rather negligible given the previous error for the neural network's prediction,

and in the interest of time, the remaining cases were carried out with the original grid size. The resulting drag curves from the ELAFINT3D code at Mach numbers ranging from 1.1 to 2.6 are shown in Figure 21.

The ANN was trained using this data set and the same number of neurons and number of iterations were used. The same order of computational time was observed as in the post-shocked flow calculations. The prediction curve of the neural network as well as the calculated transient drag curve is displayed in Figure 22.

The neural network was capable of matching the curve even into the negative force domain. The negative drag force arises when the incident normal shock traverses to the rear of the cylinder and a reflected bow shock has formed at the front of the cylinder, which leads to a higher pressure at the rear for a short period of time. However, in this case, the peak value of the drag was underestimated by the neural network. The cause of this is due to the neural network's activation function, and the summation of which is fitting a series of sigmoid functions to the curve. With data evenly distributed, a small number of data points exist near the peak. The unbalanced set causes the neural network to spend more time fitting to the rest of the curve than the peak. Another reason is that the neural network is attempting to fit with a global array, thus the overall prediction curve will be similar to a smoothing function and reduce peaks. The sharper the peak, the less likely the neural network will capture an accurate depiction. For a moving particle these sharper peaks do occur. Several solutions including the use of wavelet basis functions, neural network expansion, multi-resolution and segmentation exist; these will be discussed in detail later.

MOVING PARTICLE

For the moving particle problem, the boundary conditions, the initial conditions, domain size, and particle size remained unchanged from the previous experiment. The chosen Mach numbers allow for easier comparison to conditions used in various experiments[1, 19]. The artificial neural network was set up to segment the drag curves in time to facilitate more customized fitting in the respective segments. This would allow for a better fit to the drag curve. The training data provided to the artificial neural network is shown in Figure 23.

The total training time for the neural network was still under 30 seconds because the amount of data per iteration for each partition of the neural network was reduced. The root mean square error was significantly reduced and was less than 0.5% for 700 data points in the later time section. The resulting prediction output was also segmented according to which partition of the artificial neural network was responsible for learning the curve characteristics of the function. The resulting 40 neuron, partitioned artificial neural network produced a remarkably good prediction as shown in Figure 24.

MULTIPLE PARTICLE CASES

The drag versus time curve for a single particle is fairly easily predicted by an artificial neural network with only one interacting shock wave. It may be necessary to implement another method of data assimilation to describe more complicated functions and drag curves. The previous

experiments grew in difficulty to examine the different properties of supersonic fluid flow around a cylindrical particle. From the post-shocked experiment, the neural network observed that the drag increased as the Mach number increased and the drag went down over time. The drag force of the stationary particle displayed negative values. With the cylinder moving, the drag peaks became more prominent. A single neural network is able to derive the drag correlations from numerical methods given a single particle. When there is particle laden flow field, a new approach is needed to extract the drag correlations.

In order to obtain a general drag curve with characteristics that could be applied to any particle embedded in a cloud, there needed to be data obtained from many particles in many possible arrangements. The best way to obtain data like this was to run simulations of randomly seeded clouds and to define a “representative particle (RP)” embedded in the flow; much as in the case of “representative elementary volumes” (RVEs) employed in volume-averaged formulations of multiphase flows. One way to define such representative particles is to locate them at the center of a cloud of particles; this avoids edge effects and wave reflections from domain boundaries. The representative particles for one particular case are illustrated by the outline in Figure 30. To ensure the proper tracking of the same centralized particles, a particle array was first formed and then the particle centers were perturbed. The boundary conditions were set to simulate a shock tube for comparison to the works Boiko et al.[1], Tanno et al.[19] and Sun et al.[28]. The left edge of the domain was set as an inlet, the right edge an outlet, and both the top and bottom edges were set as reflective boundaries. An example of the flow can be seen in Figure 31.

The particles in this case number 41, each are seeded in a respective location where a 4 by 4 grid of 16 particles is embedded in 5 by 5 grid of 25 particles as seen in Figure 14. This enabled the users to easily code the location of each particle, yet create an array where every particle is staggered off the one directly in front. The slight randomization completed the task of attempting to simulate a random dispersal of particles while still being able to easily track a few. The few that were important enough were the particles embedded directly in the center of the array. The center particles experience a much more randomized collision of reflected shocks by the few rows and columns of particles behind and to each side. The drag curves for these particles were extracted by the integration of pressure over the level set boundary. The drag curves of 5 particles from the center of the cloud were then averaged. The results of the averaging of the drag curves for the RPs can be seen as the bold curve in Figure 32.

Apart from the Mach number, the other parameters that can affect the behavior of particles in a cloud include the volume fraction of particles, the particle density relative to the fluid, particle shape, collisions between particles and viscous effects as controlled by the Reynolds number. The last three effects are not considered in this work as they are expected to have secondary effects in the initial phase of shock-particle interactions. Of the three parameters considered, namely Mach number (M), particle density ratio ($\frac{\rho_p}{\rho_f}$) and volume fraction ϕ_p , the effects of the ϕ_p variable are much more easily verified by direct viewing of the flow field. Upon comparison of the shape of the incident shock already passed over the particle arrays in Figure 31 and Figure 33, there is a very definitive concavity resulting from the passage over the more dense cloud. Such an obvious

change (due to the higher impedance to shock propagation presented by the denser cloud) in the flow field would imply an equally large effect in drag force and thus the motion of the particles inside the cloud. Even though all other parameters were set the same, the dense cloud case in Figure 33 depicts more of a compressing of the cloud along the direction of flow.

Volume fraction, φ_p , is an important parameter that is different throughout any dust cloud and changes over time. It is only one of a few parameters that we chose to vary that we deemed to have the largest affect on particle motion. A comparison of the averaged drag curves (for the representative particle) for varied φ_p can be seen in Figure 34.

It may be deduced that the increase of φ_p decreases the impulse I_i delivered by the shock on a particle. The $C_{d_{max}}$ force experienced remains fairly constant but the decay of the force diminishes much more rapidly. This is due to the decreases in strength of the shock waves impinging on the RPs in the center of the cloud. Another more obvious variable that affects the drag force felt by shocked particles is the Mach number. In Figure 35 the effect of the Mach number is very obvious. As the Mach number increases, the $C_{d_{max}}$ felt rises dramatically.

This dramatic rise is directly correlated to the shock strength in the high Mach number flow. It is the flow velocity that in turns defines the Reynolds number that most fluid codes based on the non-dimensional parameters. This is why so many drag laws depend on the Reynolds number, but the Mach number is a far more relevant parameter in the initial shock-particle interaction phase. It has already been shown that the main separating factor between Reynolds and Mach number is viscosity. It has also been shown that viscosity does not affect the drag force on a particle this early. This is why we believe that developing the variation of the drag on an RP with respect to the Mach number in an inviscid model is a better parameter for lifting drag information from the meso-scale to the macro-scale.

The next parameter examined in our experiments was that of the mass of the particle. In terms of non-dimensional variables, this correlates to the mass of the particle as a ratio of the density of the solid particle and the fluid surrounding it. Most of the experimental models of shock-particle interactions employed spheres made of acrylic and bronze [1]. The medium used was air, and because of those models we chose to set $\frac{\rho_p}{\rho_f}$ near 1000. To encapsulate motion a little easier, we mainly varied $\frac{\rho_p}{\rho_f}$ lower. The comparison of drag forces are displayed in Figure 36. The data obtained shows correlations per each variable, the ANN will hopefully connect them together and engage its application to multiscale modeling.

5. APPLICATION OF ANN-BASED LEARNING TO MULTI-SCALE COMPUTATIONS

INFORMATION PASSAGE

To utilize the correlations obtained previously to use in multi-scale modeling, information must be lifted from the meso-scale. The transient drag curve is quite a lot of information to pass between scaling levels in multiscale modeling. With regard to the drag curves acquired, there were two important parameters needed for particle motion. Both Kosinska[44] and Kosinski[3]

showed that the linear motion of a rigid body even those immersed in shock waves can be found directly from Newton's second law and derivations of it. Newton's second law directly correlates force to mass and acceleration. To determine the speed and position we would need to know the momentum transferred and the rate of momentum transferred. The momentum and rate of transfer can be found as an expression of I_t and τ_r .

From Equation 11, we have a simple method of determining the total impulse, I_t , a particle would experience over time. Integrating Equation 11 from instant of shock impact to long times (when the inertia delivered by the shock has equilibrated particle motion, but still short enough that viscous effects can be neglected) results in:

$$I_t = C_{d_{max}} * \tau_r \text{ Equation 12}$$

The maximum drag, $C_{d_{max}}$, is easily acquired, thus the next step be to fit the relaxation time, τ_r , to the drag curve of each case the ANN will learn from for Mach number, $\frac{\rho_p}{\rho_f}$ and ϕ_p .

SINGLE PARTICLE MOTION

For each case presented, the quantified values for particle motion, $C_{d_{max}}$ and τ_r to attain I_t were found. The value of I_t and τ_r were found by numerical integration and fitting an exponential decay function by minimizing the error between the drag curve and the exponential function. One such fitting with the impulse highlighted can be seen in Figure 37.

We began with the data from our single particle cases because the drag curve fitting was simpler and straight forward without large errors due to the oscillations in drag. From that data we fed it into the ANN and obtained a hyper-surface to incorporate each variable. The 3 dimensional breakdown between two of the variables and there target parameter can be seen in the plots of Figure 38 and Figure 39

It has already been determined, and one can see from the plots, that both the I_t and τ_r increase with Mach number. This has been shown many times before by other researchers. [19, 34]^[28] It is interesting that the value of I_t actually gets steeper as the Mach number increases, making it a high order relationship. It is also important to note that both I_t and τ_r seem to approach zero near Mach 1. One can accredit that to the dramatic decrease in drag once relative velocity falls below the supersonic range. As for the effect that $\frac{\rho_p}{\rho_f}$ has, both I_t and τ_r level off toward higher values, as $\frac{\rho_p}{\rho_f}$ approaches the representation of an infinitely massive or stationary particle. For when $\frac{\rho_p}{\rho_f}$ approaches zero, both I_t and τ_r approach zero as a very small particle's motion should nearly behave the same as the fluid.

GENERAL PARTICLE MOTION

It is easy to see how a single particle would behave, but with multiple particles there occurs many complex reflective shock waves. To ensure that the general behavior of shocked particles is accurately learned by a neural network, data needs to be collected from many particles in a

random orientation. Therefore the effect of a specific array setup would be diluted and more general values could be collected. The values of $C_{d_{max}}$ force and τ_r are still the two most important parameters that can be directly obtained from the ELAFINT3D code. For particle motion that occurs in a dusty gas, another input parameter should be taken into consideration. The value of φ_p plays a particularly important part in shock-impacted particle laden flows to learn how much the $C_{d_{max}}$, τ_r and I_t is affected by the φ_p in a multiple particle cloud, 45 different cases were performed using initial values recorded in Table 3. Each case had 41 particles placed in a staggered array and then randomly perturbed to simulate a dusty gas while still being set at a standard interval to better capture the affects of φ_p .

The ANN was trained twice, once for $C_{d_{max}}$ and once for τ_r . The value of I_t is implied by the application of these two variables. The training period lasted for 5000 iterations with 25 neurons and the convergence curve is seen in Figure 40. The relationship of Mach number and φ_p versus $C_{d_{max}}$ can be seen in Figure 41, Mach number and φ_p versus τ_r , in Figure 42, and Mach number and φ_p versus I_t in Figure 43.

It becomes obvious that the major contributor for $C_{d_{max}}$ is the Mach number. The φ_p does not seem to affect $C_{d_{max}}$ at low Mach numbers. As for τ_r , both Mach number and φ_p have great affects. At low Mach numbers, the τ_r greatly increases. Drag force at subsonic velocities are relatively slow to apply. The φ_p has a major affect only at low Mach numbers. At higher Mach numbers the effect of φ_p goes away, soon thereafter, one may assume that the particles are no longer going to be shielded by particles, but actually hit by them. The surface trend for I_t is somewhat expected, the general trend being that I_t increases as the φ_p decreases and Mach number increases. For the averaged data for all cases, refer to Table 4 in the appendix.

Of course there exist errors in our model that arise from many areas, for example the averaging of multiple individually shocked particles. Very little error was displayed in the single particle cases because there was no random particle-shock interaction from reflections. Testing consisted of randomly selecting a single data point and removing it from the training set. The ANN would be reset and learn the new training set without the point being tested. The ANN was then queued at the test point and was then checked for error. The testing phase consisted of testing a few points by this method; with only one point missing on a multidimensional map, visualization is difficult to show. Testing by selection and removal showed errors all under 2%. For the multiple particle cases, errors ranged greatly. During the training phase of the multiple particle cases, the average error for the training data was less than 1%. However, due to the unsteady curvature and some areas of inconsistent trends, the average error for the prediction of randomly removed and tested points inside the ANN prediction curve for I_t were 7.3%. The largest error for the tested cases resulted from the Mach 4.4 cases which are also responsible for the extra bump on the plots of $C_{d_{max}}$ and I_t . When cases where the Mach number was 4.0 or above was left out and tested for, errors between 12.2% and 14.6% would occur.

LAGRANGIAN ADVECTION

Now that we have a trained ANN with the correlation of Mach number, $\frac{\rho_p}{\rho_f}$ and φ_p to I_t on a particle, we can use it to predict how a shock impacted particle will move. Restating what was mentioned before, one may use the $C_{d_{max}}$ and τ_r to recreate a drag curve represented by exponential decay from the $C_{d_{max}}$ at the “point” of impact. With a defined drag curve, the trajectory of a particle can be predicted by simple Lagrangian advection using Newton’s first law of motion. We performed this advection scheme with case data to match previous experiments of Boiko. Our data was limiting to simulating values of φ_p down to 2.0 percent due to the constraints of domain size, particles placed ever 15 diameters away would have produced over 5 million grid cells. The result of using data from the ANN and this Lagrangian advection scheme can be seen as the solid line alongside the experimental work of Boiko et al.[1] in Figure 44.

The symbols are directly from experimentation, the dashed line is Boiko’s computation, and the solid line is our Lagrangian advection using lifted behavior learned by the ANN. To insure the proper values of I_t were used, an numerical integration scheme was used on our data. This lead to a slightly higher initial peak and exponential representation tends to decay a slightly faster than normal as seen in Figure 37. The largest error is near the beginning where the initial impact of our model is piecewise and thus sharper. However, the ANN and Lagrangian advection model is a close representation of how a particle moves.

MACRO-SCALE PHENOMENA

Now that we are able to predict movement of a particle using data from the meso-scale, we should be able to adapt that to multiple particles at the macro-scale. With the ELAFINT3D code running on a serial machine with limited random access memory, we are currently limited to particle clouds of less than 180 particles. To maintain the same staggered and perturbed setup, we chose to model a cloud using 145 particles. This particular case ran with a domain size of 60 by 70 d_p and the smallest cells having a grid size of 0.03 d_p approaching 4 million cells and thus reaching the limit of memory on the machine. To arrive at this point, the model ran more than 25 non-dimensional units of time for about a wall clock time of three weeks. A Schlieren image of this case can be seen in Figure 45. We were able to then use the drag data from each particle to advect their location further. The result was merely the compression of the cloud moved a few domains along the flow direction.

There are no experimental demonstrations of this type of cloud; however, in the experimental work of Boiko et al. [1] a macro-scale phenomenon of larger and denser dusty gas clouds emerges. Referring once again to Figure 2 (Chapter 1), one can observe the formation of a sideways “V”. This is a phenomenon that arises only in the cases where the dust clouds are sufficiently dense. It is not observable Figure 1 or in the other cases performed by Boiko et al.[1] as seen in Figure 46. In each of the cases presented, a thin band of particles is shocked.

In any case, the Mach number and the $\frac{\rho_p}{\rho_f}$ remains virtually the same throughout the whole domain, yet the particles obviously move at different velocities, which mean they would have different values of I_t . The first two factors that arise that may affect I_t are φ_p and the shielding of

the shock wave by particles ahead in the flow domain. Shielding is of course directly related to φ_p as well as the total number of particle in the domain. Since the total number of particles in a domain is an extrinsic property of a dust cloud, it is not advisable to use it. This would have to be utilized via a decrease in Mach number. Another case was performed in where a triangular dust cloud was shocked to observe the effect of shielding. Both this case and others found that within a certain φ_p , the total number of particles does not drastically affect particle motion.[1]^[45-46] Thus the main contributing factor to the variance in particle motion in a single domain with constant Mach number and $\frac{\rho_p}{\rho_f}$, is φ_p . Knowing this, a much larger simulation can be performed with drag forces obtained from an ANN which learned from cases with varying φ_p .

MACRO-SCALE SIMULATION

For the macro-scale simulation, we used a Lagrangian advection scheme to move particles based on the drag force obtained from $C_{d_{max}}$ and τ_r predicted by an ANN given Mach number, $\frac{\rho_p}{\rho_f}$ and φ_p . The Mach number and $\frac{\rho_p}{\rho_f}$ was predefined while the φ_p was calculated based on the area fraction (in 2D) occupied by the particles, computed for a box of 20 by 20 diameters in the level set field with a domain size of 1024 by 512 grid points as seen in Figure 47, Figure 48 and Figure 49.

ASSUMPTIONS

To aid in modeling the shock-impacted particle-laden flow, certain assumptions were made. The assumptions and their implications are as follows:

Particles are normally distributed in both the x and y directions for dust clouds. They are completely still at beginning with no velocity components.

All small scale forces are neglected. This ignores the effects of buoyancy, gravity, electromagnetic forces, chemical attraction, or Brownian motion.

Particles are treated as points. No collisions occur in calculations and particles may over lap or pass through each other. To stimulate y direction and small forces, a random perturbation of location was included.

φ_p is computed by summing the surrounding first level set, particle volumes overlapping are neglected.

$C_{d_{max}}$ force and τ_r computed by ANN in the first step, thus this is as if the incident shock impacted every particle at the same time

Largest variable factor in drag is φ_p , particles with very high φ_p barely move, due to lack of collisions.

For an example of the algorithm, a pseudo code has been provided in the attachments.

GENERAL MOTION

To ascertain that indeed the formation of the “V” shaped phenomenon is due to that of the variation in φ_p several macro-scale models were performed. They included simulations that were drag law based, with low φ_p , with high φ_p , and with a uniform band φ_p . For the case based on a drag law, the “standard” drag law found in Table 2 was used to determine the force on each particle. This straightforward method and the assumptions made above caused every point to move roughly the same amount as seen in Figure 50.

For the sparse dust cloud case, Figure 51, similar actions occurred due to a small variance in φ_p . Demonstrated experimentally, little difference in movement also occurs in Figure 1, of Boiko’s experiments. With φ_p and other parameters all the same, each particle should experience the same motion. When the density of particle is increased such as in Figure 52, a “V” phenomenon would appear as seen in Figure 2 by Boiko et al.[1] The formation of this phenomenon occurs only at the macro-scale when there is a wide range in φ_p . In Boiko’s experiment one can observe a block of particles in the middle. Just a block alone is not capable of producing a strong enough variance in particle velocity to form a “V”. The simulation in Figure 45 demonstrated no large differences in I_t or velocity. When a band of particles was used, as in Figure 53, more particles were spread out just behind the cloud. This is most obvious in Figure 53d and Figure 46d frame 2 where the left side of the block is evidently denser than the right.

6. CONCLUSIONS

The objective of this thesis was to efficiently model the interaction of a shock wave and a dusty gas. We wanted to accomplish this by formulating an algorithm to learn the behavior of meso-scale simulations. We successfully set up and used a feed-forward back propagation artificial neural network to learn the drag curves from single and multiple particle cases. For application to multiscale modeling, we chose important characteristics from the meso-scale simulations to be “lifted” into a macro-scale simulation. The values of $C_{d_{max}}$ and τ_r formulated a transient representation of I_t . The values of I_t learned using the ANN had an average error of less than 0.5% in training and 2.0% in testing for single particles and less than 1.0% and 8.0% for multiple particles. The multiple particle cases provided more variance in the data of each particle separately than the variance of the ANN learning. The learned behaviors by the ANN were used in macro-scale simulations. The different macro-scale simulations demonstrated the great improvement of using an ANN and multiscale methods over traditional methods using predefined drag laws.

The largest simulation that was performed for this thesis was that of a shock impacted 145 particle cloud. This simulation took a wall clock time of nearly 3 weeks on a serial processor and used 16 gigabytes of random access memory prior to stalling out near 30 non-dimensional units of time between 3 million and 4 million cells. To run a simulation with 1000 particles in a domain nearly 125 times larger it would take the same machine, assuming limitless memory, roughly 50 years. The ANN learned behavior in the macro-scale simulation performed such a task in less than one minute. The data processed was 45 cases each lasting a few days but is capable of running in parallel. In either case, with the use of behavioral learning of ‘lifted’ meso-scale

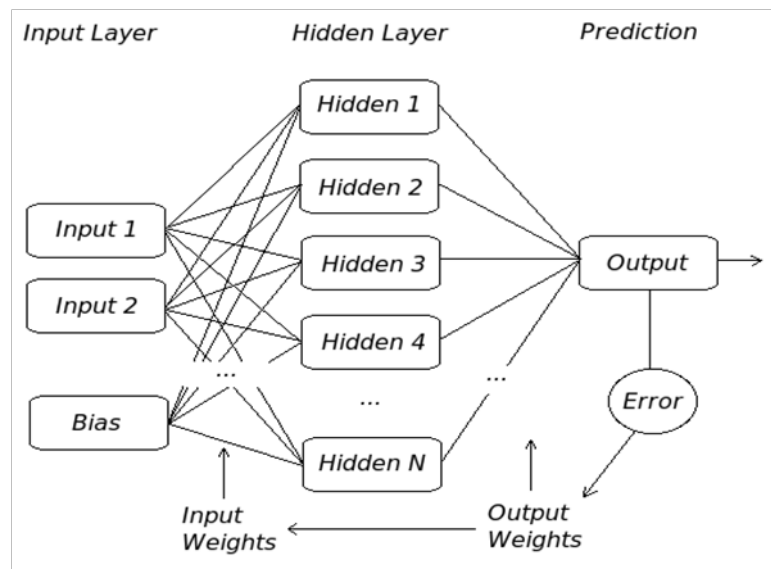
variables, much time was saved. Disregarding a lifting of single variables, the ANN could still learn the entirety of the drag curve for complicated scenarios if aided by multi-resolution analysis.

7. REFERENCES

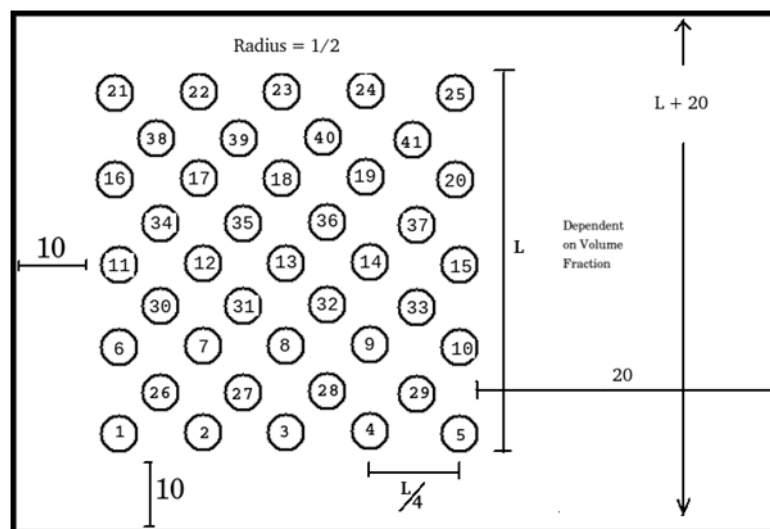
1. Boiko, V.M., Kiselev, V.P., Kiselev, S.P., Papyrin, A.N., Poplavsky, S.V., Fomin, V.M., Shockwave Interaction with a Cloud of Particles. *ShockWaves*, 1997. 7: p. 275-285.
2. Crowe, C.T., Sommerfeld, M. and Tsuji, Y., *Multiphase flows with droplets and particles*. 1988.
3. Kosinski, P., On Shock Wave Propagation in a Branched Channel with Particles. *Shock Waves*, 2006. 15(1): p. 13-20.
4. Kosinski, P., Numerical analysis of shock wave interaction with a cloud of particles in a channel with bends. *International Journal of Heat and Fluid Flow*, 2007. 28(5): p. 1136-1143.
5. Fedorov, A.V., Y.V. Kharlamova, and T.A. Khmel, Reflection of a shock wave in a dusty cloud. *Combustion Explosion and Shock Waves*, 2007. 43(1): p. 104-113.
6. Fausett, L.V., *Fundamentals of Neural Networks*, ed. D. Fowley. 1994, Upper Saddle River, New Jersey, United States: Prentice-Hall.
7. Nilsson, J., M. Ohlsson, P. Hoglund, B. Ekmehag, and B. Koul, Artificial Neural Networks - A Method for Optimal. Donor-Recipient Matching. Large Scale Simulation of Survival after Heart Transplantation. *Journal of Heart and Lung Transplantation*, 2010. 29(2): p. S29-S29.
8. Krose, B. and P.v.d. Smagt, *An introduction to Neural Networks*. 1996, Amsterdam, Amsterdam: The University of Amsterdam.
9. Unger, J.F. and C. Konke, Coupling of scales in a multiscale simulation using neural networks. *Computers & Structures*, 2008. 86(21-22): p. 1994-2003.
10. Jorg F. Unger, C.K., Coupling of scales in a multiscale simulation using neural networks. *Computers & Structures*, 2008. 86(21): p. 1994-2003.
11. Ahmadi, M., M. Saemi, and K. Asghari, Estimation of the reservoir permeability by petrophysical information using intelligent systems. *Petroleum Science and Technology*, 2008. 26(14): p. 1656-1667.
12. Giralt, F., Arenas, A., Ferre-Gine, J., Rallo, R., The simulation and interpretation of free turbulence with a cognitive neural system. *Physics of Fluids*, 2000. 12(7): p. 1826.
13. Sahimi, M., Fractal-wavelet neural-network approach to characterization and upscaling of fractured reservoirs. *Computers & Geosciences*, 2000. 26(8): p. 877-905.
14. Unger, J.F. and C. Konke, Neural networks as material models within a multiscale approach. *Computers & Structures*, 2009. 87(19-20): p. 1177-1186.
15. Ghaboussi, J., D.A. Pecknold, M.F. Zhang, and R.M. Haj-Ali, Autoprogressive training of neural network constitutive models. *International Journal for Numerical Methods in Engineering*, 1998. 42(1): p. 105-126.
16. Hocevar, M., Sirok, B., Grabec, I., A Turbulent-Wake estimation using radial basis function neural networks. *Flow, Turbulence and Combustion*, 2005. 74: p. 291-308.

17. Bishop, C.M., Neural Networks for Pattern Recognition. 1996, USA: Oxford University Press.
18. Mehrotra, K., Mohan, C.K., Ranka, S., Elements of Artificial Neural Networks. 1996, Cambridge: Massachusetts Institute of Technology.
19. Tanno, H., Itoh, K., Saito, T., Abe, A., Takayama, K., Interaction of a Shock Wave with a Sphere Suspended in a Vertical Tube. *Shock Waves*, 2003. 13(3): p. 191-200.
20. Chen, L., Debenedetti, P.G., Gear, C.W., Kevrekidis, I.G., From Molecular Dynamics to Coarse Self-Similar Solutions: A Simple Example Using Equation-Free Computation. *Journal of Non-Newtonian Fluid Mechanics*, 2004. 120(1): p. 215-223.
21. Kevrekidis, I.G., Gear, C.W., Hummer, G., Equation-Free: The Computer-Aided Analysis of Complex Multiscale Systems. *Aiche Journal*, 2004. 50(7): p. 1346-1355.
22. Samaey, G., I.G. Kevrekidis, and D. Roose, Patch dynamics with buffers for homogenization problems. *Journal of Computational Physics*, 2006. 213(1): p. 264-287.
23. Samaey, G., D. Roose, and I.G. Kevrekidis, The gap-tooth scheme for homogenization problems. *Multiscale Modeling & Simulation*, 2005. 4(1): p. 278-306.
24. Gear, C.W., J. Li, and I.G. Kevrekidis, The gap-tooth method in particle simulations. *Physics Letters A*, 2003. 316(3-4): p. 190-195.
25. Igra, O., Takayama, K., Shock Tube Study for the Drag Coefficient of a Sphere in a Non-Stationary Flow. *Proc. R. Soc. Lond.*, 1993. 442(A): p. 231-247.
26. Rogak, S.N. and R.C. Flagan, Stokes Drag on Self-Similar Clusters of Spheres. *Journal of Colloid and Interface Science*, 1990. 134(1): p. 206-218.
27. Saito, T., M. Saba, M. Sun, and K. Takayama, The effect of an unsteady drag force on the structure of a non-equilibrium region behind a shock wave in a gas-particle mixture. *Shock Waves*, 2007. 17(4): p. 255-262.
28. Sun, M., Saito, T., Takayama, K., Tanno, H., Unsteady Drag on a Sphere by Shock Wave Loading. *Shock Waves*, 2005. 14(1): p. 3-9.
29. Tanno, H., K. Itoh, T. Saito, A. Abe, and K. Takayama, Interaction of a shock with a sphere suspended in a vertical shock tube. *Shock Waves*, 2003. 13(3): p. 191-200.
30. Boiko, V.M., V.P. Kiselev, S.P. Kiselev, A.N. Papyrin, S.V. Poplavsky, and V.M. Fomin, Shock wave interaction with a cloud of particles. *Shock Waves*, 1997. 7(5): p. 275-285.
31. Drikakis, D., D. Ofengeim, E. Timofeev, and P. Voionovich, Computation of non-stationary shock-wave/cylinder interaction using adaptive-grid methods. *Journal of Fluids and Structures*, 1997. 11(6): p. 665-691.
32. Fedorov, A.V., Kharlamova, Y.V., Khmel, T.A., Reflection of a Shock Wave in a Dusty Cloud. *Combustion, Explosion, and Shock Waves*, 2007. 43(1): p. 104-113.
33. Saito, T., Numerical analysis of dusty-gas flows. *Journal of Computational Physics*, 2002. 176(1): p. 129-144.
34. Saito, T., Marumoto, M., Takayama, K., Numerical Investigations of Shock Waves in Gas-Particle Mixtures. *Shock Waves*, 2003. 13: p. 299-322.

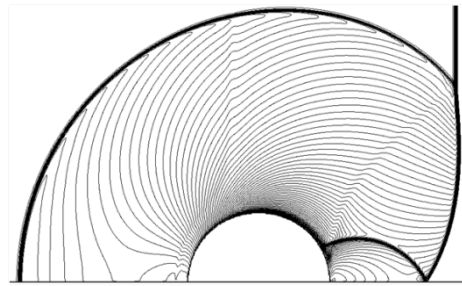
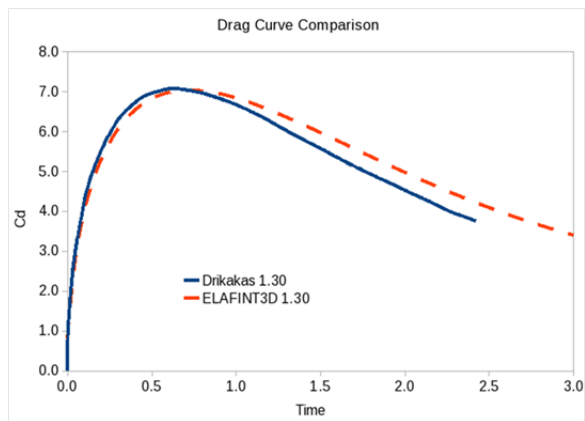
35. Saito, T., M. Marumoto, H. Yamashita, S.H.R. Hosseini, A. Nakagawa, T. Hirano, and K. Takayama, Experimental and numerical studies of underwater shock wave attenuation. *Shock Waves*, 2003. 13(2): p. 139-148.
36. Sommerfeld, M., The Unsteadiness of Shock-Waves Propagating through Gas-Particle Mixtures. *Experiments in Fluids*, 1985. 3(4): p. 197-206.
37. Khmel', T.A. and A.V. Fedorov, Interaction of a shock wave with a cloud of aluminum particles in a channel. *Combustion Explosion and Shock Waves*, 2002. 38(2): p. 206-214.
38. Igra, O. and K. Takayama, Shock-Tube Study of the Drag Coefficient of a Sphere in a Nonstationary Flow. *Proceedings of the Royal Society of London Series a-Mathematical Physical and Engineering Sciences*, 1993. 442(1915): p. 231-247.
39. Ben-Dor, G., O. Igra, and L. Wang, Shock wave reflections in dust-gas suspensions. *Journal of Fluids Engineering-Transactions of the Asme*, 2001. 123(1): p. 145-153.
40. Sambasivan, S., Udaykumar, H.S., An Evaluation of Ghost-Fluid Methods for Strong Shock Interactions with Immersed Solid Interfaces. 2009, University of Iowa: Iowa City.
41. Osher, S., Sethian, J.A., Fronts Propagating with Curvature Dependent Speed Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 1988. 79: p. 12-49.
42. Fedkiw, R.P., Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *Journal of Computational Physics*, 2002. 175(1): p. 200-224.
43. Fedkiw, R.P., T. Aslam, B. Merriman, and S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 1999. 152(2): p. 457-492.
44. Kosinska, A., Interaction of debris with a solid obstacle: Numerical analysis. *Journal of Hazardous Materials*, 2010. 177(1-3): p. 602-612.
45. Khmel, A.V., Fedorov, T.A., Interaction of a Shock Wave with a Cloud of Aluminum Particles in a Channel. *Combustion, Explosion, and Shock Waves*, 2002. 38(2): p. 206-214.
46. Ben-Dor, G., Mond, M., Igra, O., Martsiano, Y., A Nondimensional Analysis of Dusty Shock Waves in Steady Flows. *KSME Journal*, 1988. 2(1): p. 28-34.



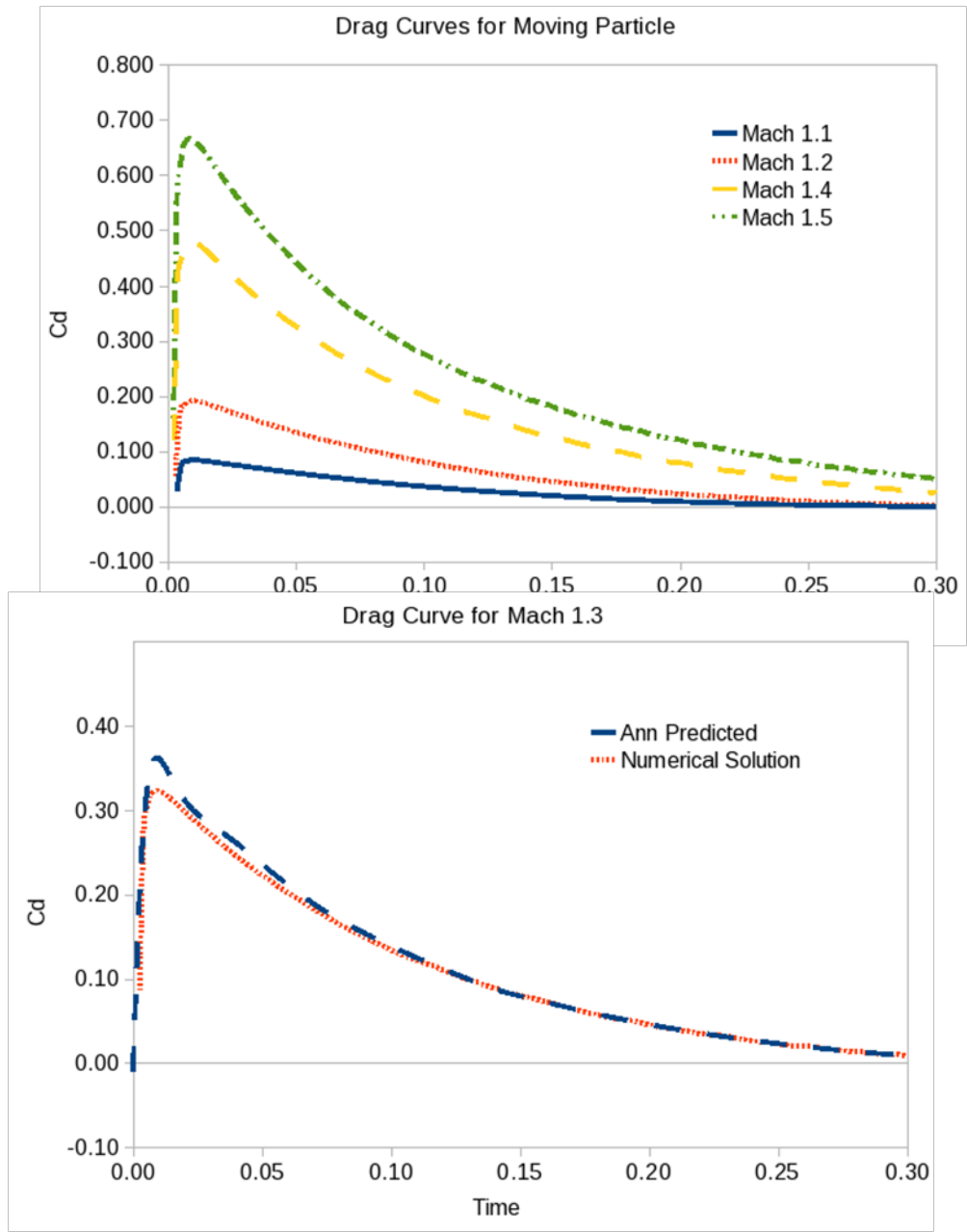
ANN topography



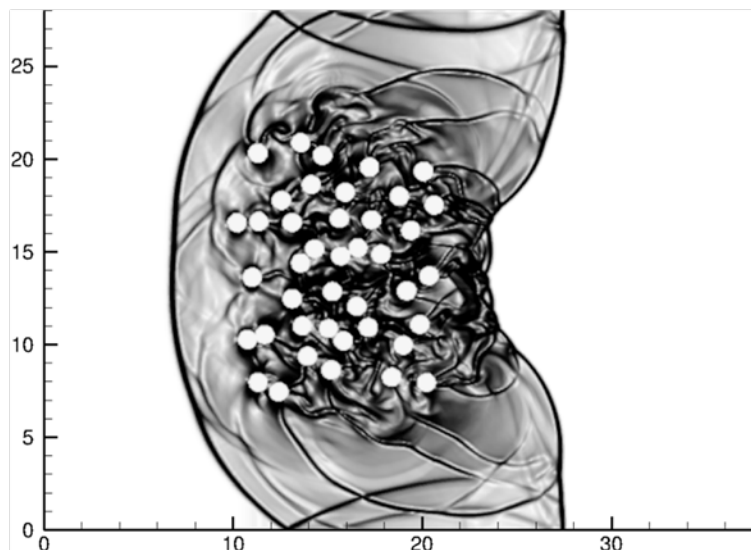
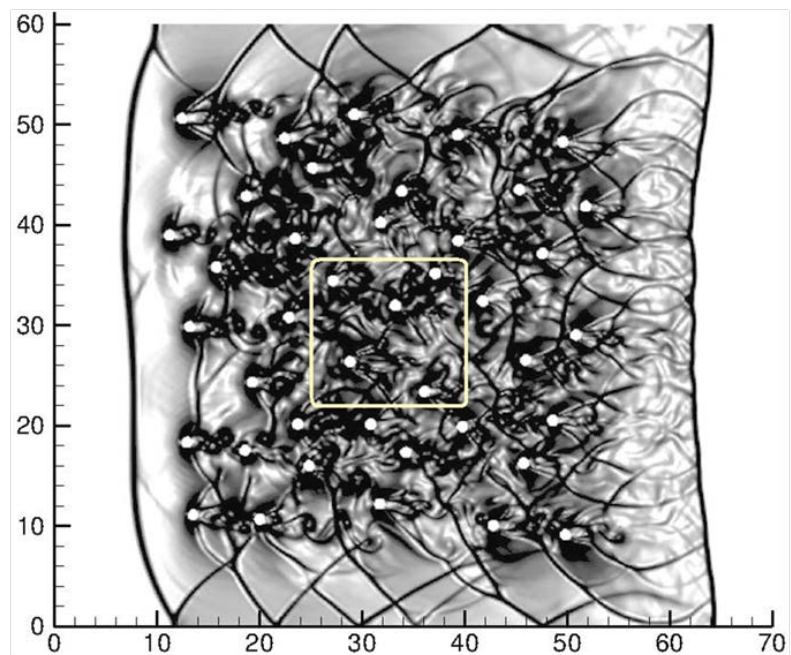
Particle array setup

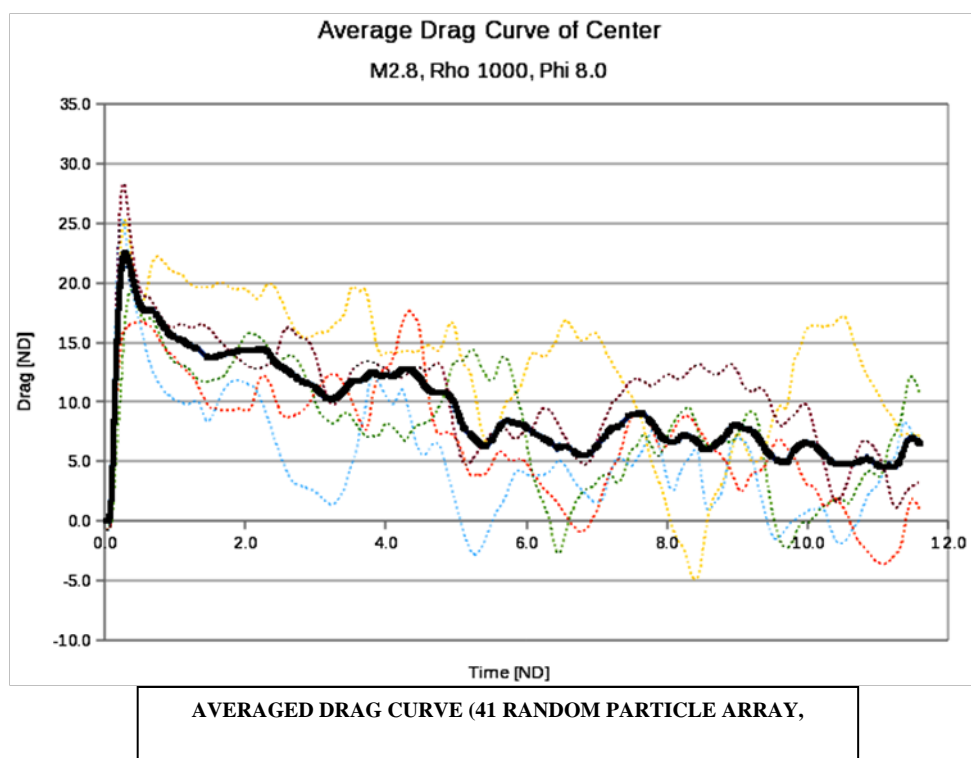


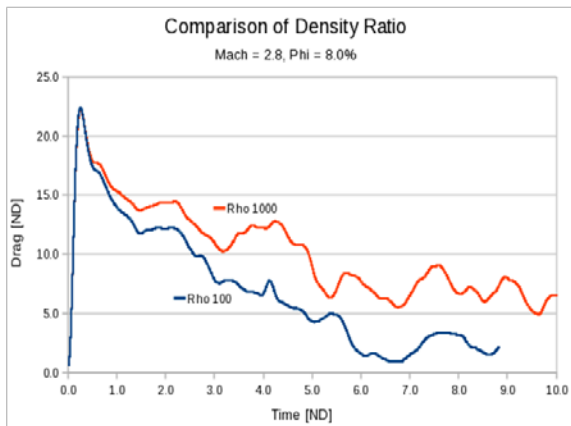
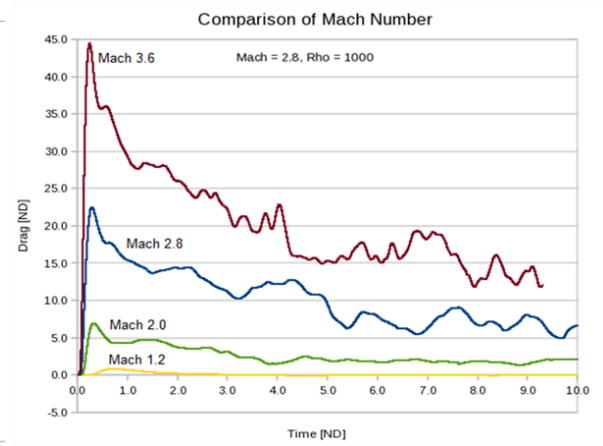
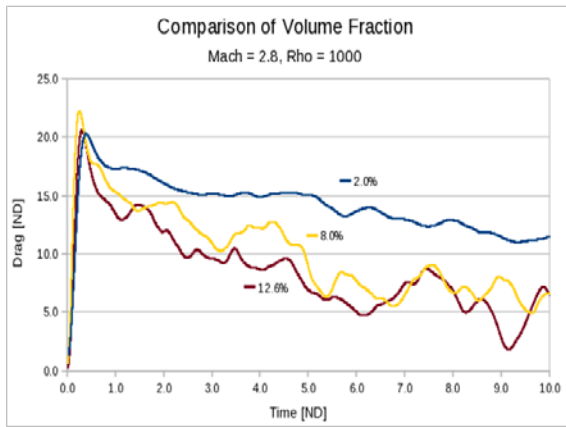
Comparison of drag vs time

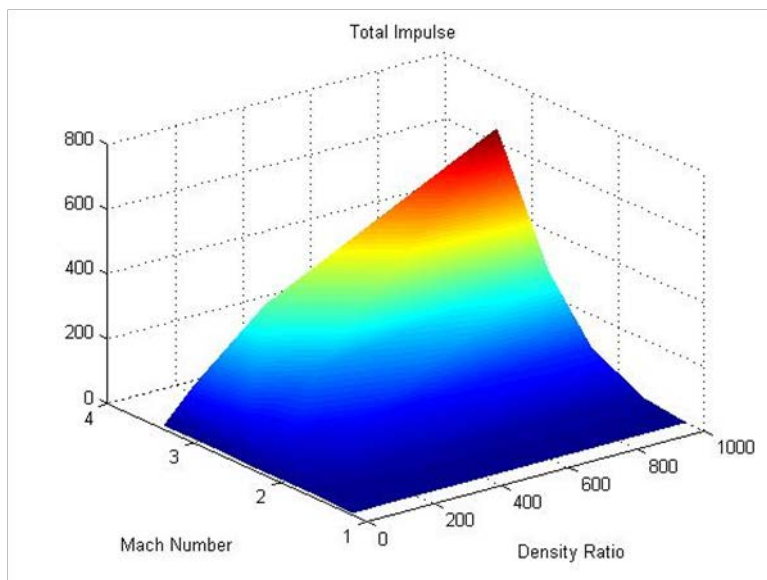
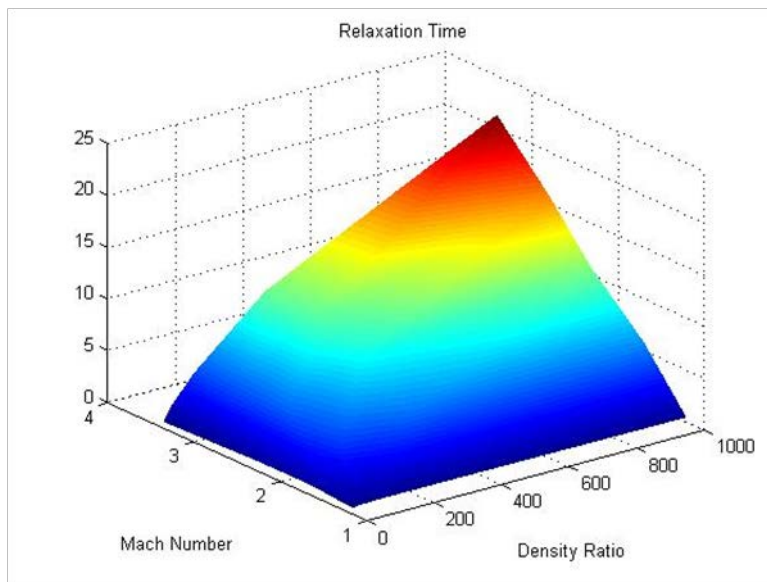


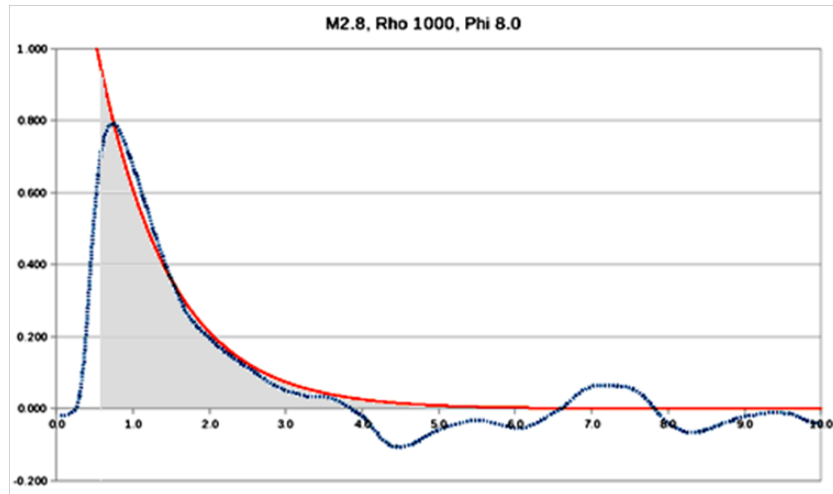
Moving particle drag curves

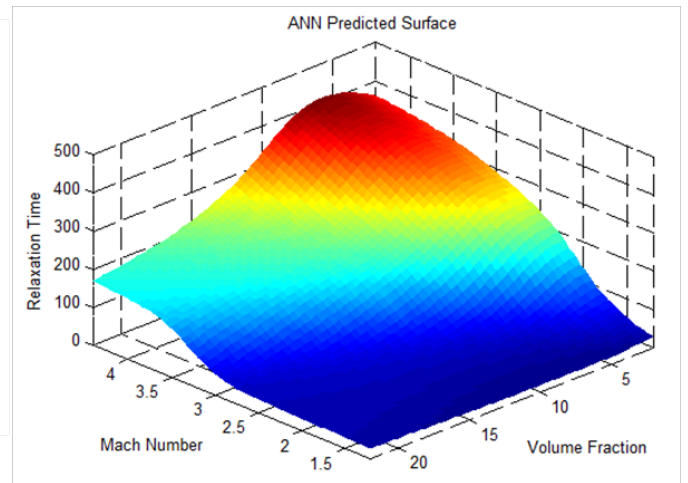
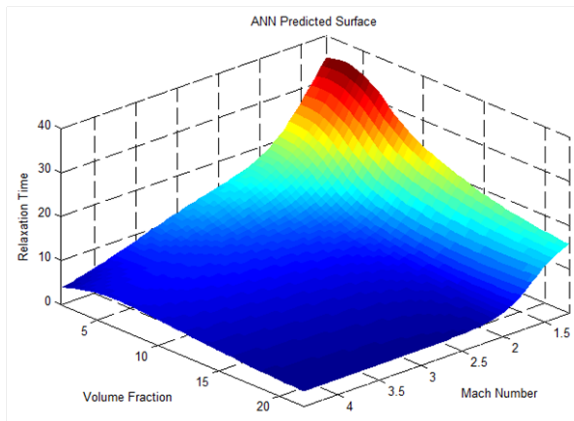
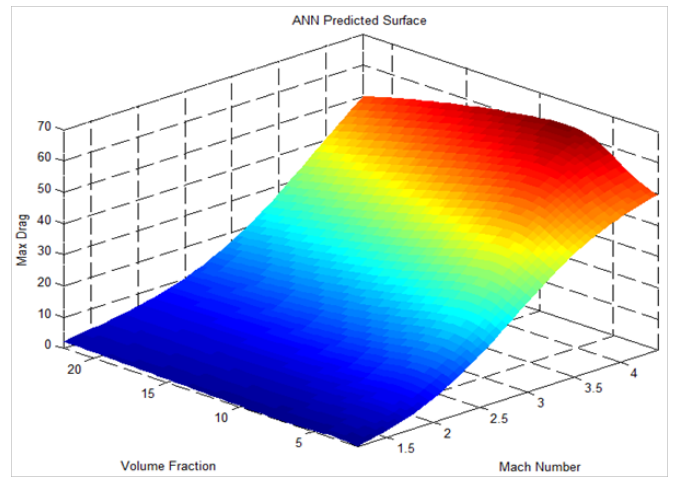
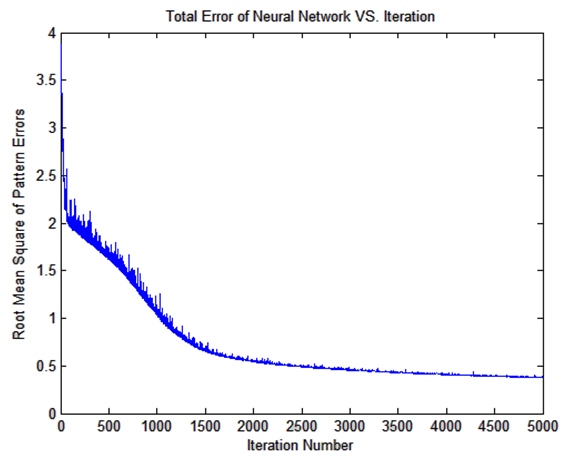


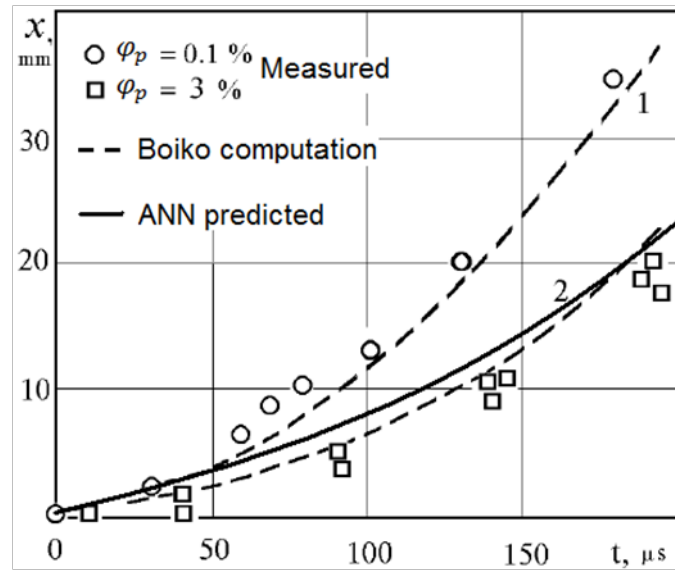


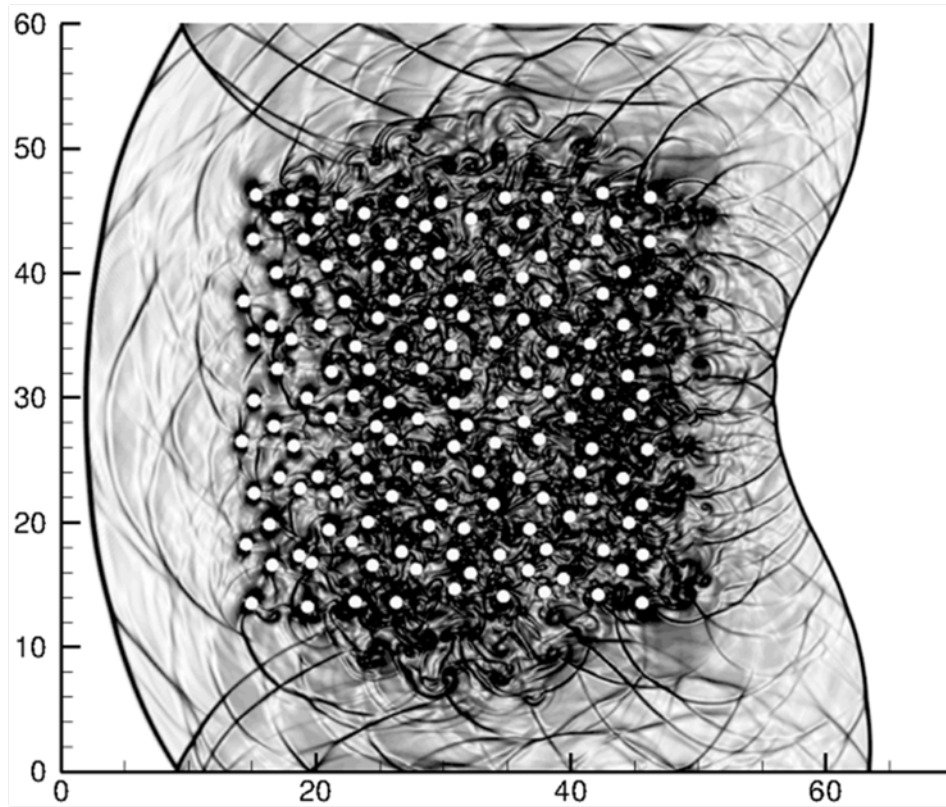


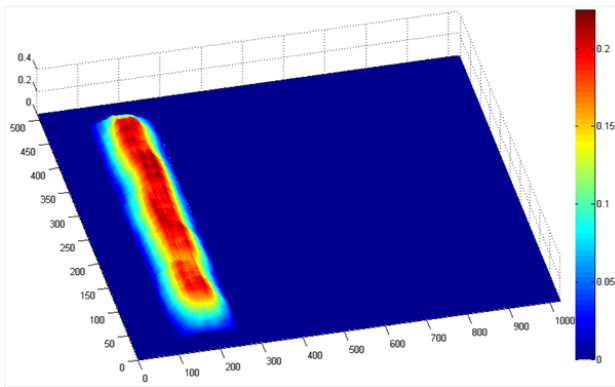
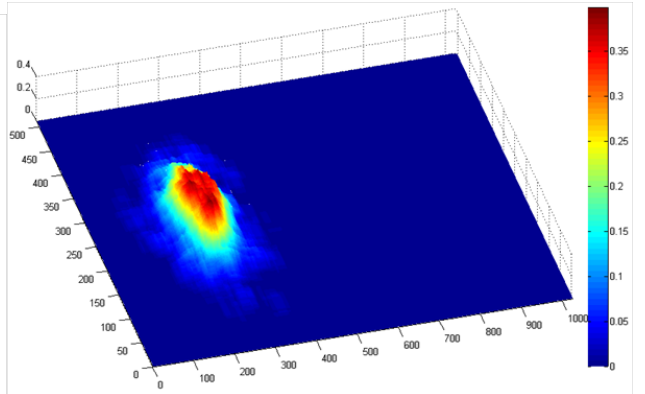
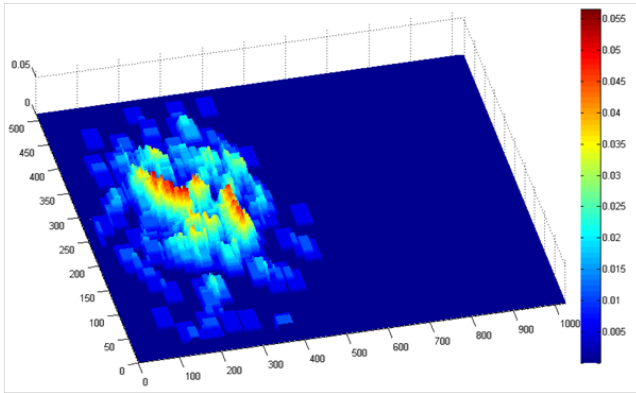


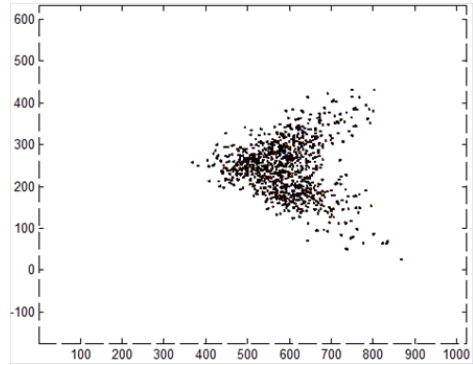
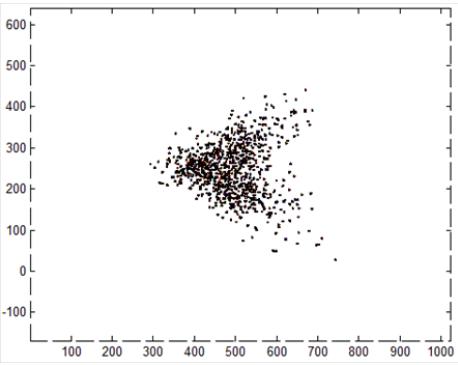
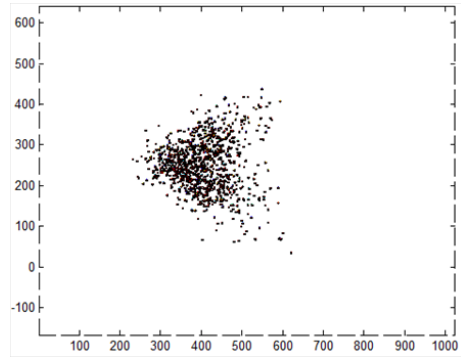
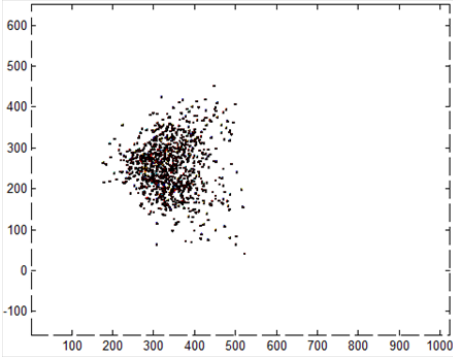
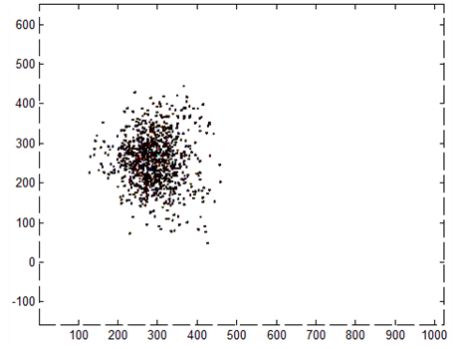
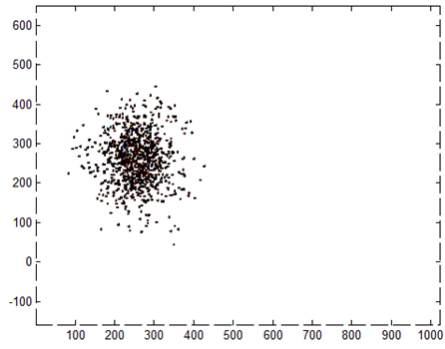


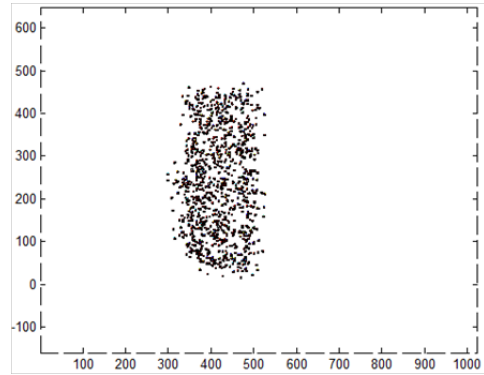
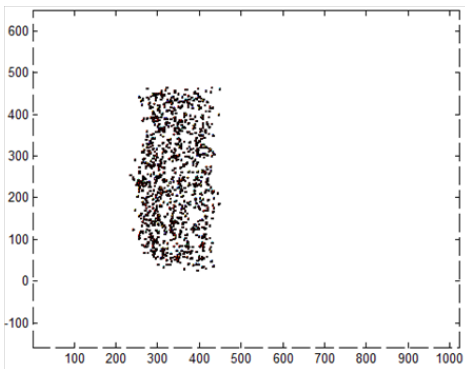
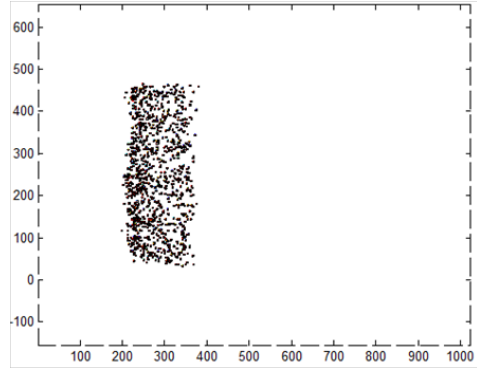
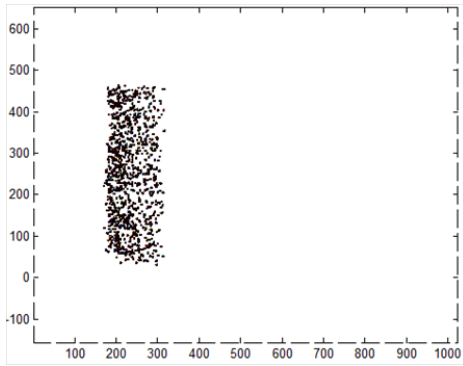
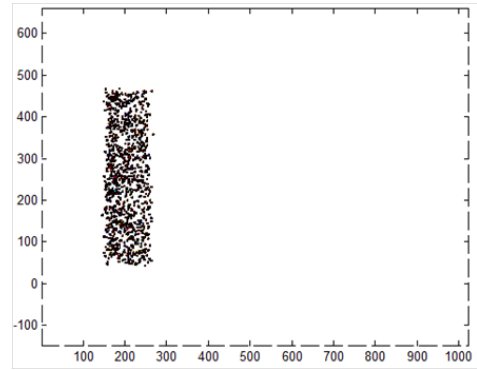
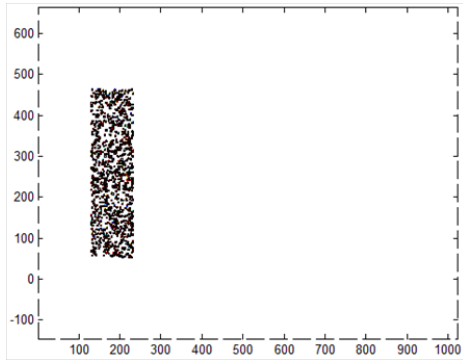












Mach	$\frac{\rho_p}{\rho_f}$	$\Phi_{\text{vol}} [\%]$	Mach	$\frac{\rho_p}{\rho_f}$	$\Phi_{\text{vol}} [\%]$	Mach	$\frac{\rho_p}{\rho_f}$	$\Phi_{\text{vol}} [\%]$
1.2	100	2.0	2.0	1000	8.0	2.8	1000	22.4
1.2	100	8.0	2.0	1000	12.6	2.8	3100	8.0
1.2	100	12.6	2.0	1000	22.4	3.2	310	8.0
1.2	310	22.4	2.4	100	8.0	3.2	310	22.4
1.2	1000	2.0	2.4	100	22.4	3.6	100	2.0
1.2	1000	8.0	2.4	310	2.0	3.6	100	8.0
1.2	1000	12.6	2.8	31	8.0	3.6	100	12.6
1.6	31	2.0	2.8	100	2.0	3.6	1000	2.0
1.6	310	8.0	2.8	100	8.0	3.6	1000	8.0
2.0	100	2.0	2.8	100	12.6	3.6	1000	22.4
2.0	100	8.0	2.8	100	22.4	4.0	100	22.4
2.0	100	12.6	2.8	310	8.0	4.0	310	2.0
2.0	100	22.4	2.8	1000	2.0	4.0	1000	8.0
2.0	310	12.6	2.8	1000	8.0	4.4	1000	8.0
2.0	1000	2.0	2.8	1000	12.6	4.4	3100	12.6